

In re Application of DURR, et al.
Application No. 09/801,951

Amendments to the Claims

1. (Currently Amended) A method for loading an executable program and associated modification code segments, the method comprising the steps of:
receiving a set of original program segments referenced by the executable program,
each of the set of original program segments comprising executable binary code;
identifying a set of substitute program segments associated with ones of the original set of program segments;
modifying executable binary code of the ones of the original set of program segments to include an exception inducing code, the exception inducing code comprising executable binary code; and
creating an association between ones of the set of original program segments and ones of the set of substitute program segments.
2. (Original) The method of claim 1 further comprising determining a condition under which a particular substitute program segment will not be executed in place of an associated original program segment.
3. (Original) The method of claim 1 wherein the substitute program segments comprise functions.
4. (Original) The method of claim 3 wherein the functions comprise application program interfaces.
5. (Original) The method of claim 4 wherein the application program interfaces are part of an operating system.
6. (Original) The method of claim 1 wherein the creating an association step comprises building a table entry including a called original program segment reference and a substitute program segment reference.

In re Application of DURR, et al.
Application No. 09/801,951

7. (Original) The method of claim 6 wherein the called original program segment reference comprises an address.

8. (Original) The method of claim 6 further comprising storing a portion of the original program segment replaced by the exception inducing code in table entry.

9. (Original) The method of claim 6 further comprising storing a condition description specifying a program execution state under which a substitute program segment will not be executed in place of an associated called original program segment.

10. (Original) The method of claim 1 wherein the identifying step comprises searching a program modification database including a set of identified programs and corresponding substitute program segments.

11. (Currently Amended) The method of claim 1 further comprising the steps of:

loading ~~the an~~ executable program into active process space;
identifying an in-memory patch within the program modification database corresponding to the executable program; and
inserting ~~the an~~ in-memory patch within the executable program residing in the active process space.

12. (Currently Amended) A method for loading an executable program comprising executable binary code and having associated modification code segments, the method comprising the steps of:

loading, for execution, a copy of the executable program in active process space;
identifying a set of modifications corresponding to the executable program; and
modifying executable binary code of the copy of the executable program in active process space to include an exception inducing code, wherein the exception inducing code comprises executable binary code that induces initiates execution of an exception handling

In re Application of DURR, et al.
Application No. 09/801,951

routine for determining and executing at least a corresponding one of the set of modifications.

13. (Currently Amended) A method for implementing execution of substitute program segments associated with an active executable program, wherein the active executable program has been modified to include ^{name} exception inducing code that induces an exception handled by an exception handling routine, the method comprising the steps of:

detecting an instance of ^{name} the exception inducing code;

first determining a corresponding substitute program segment entry in a program segment substitution list; and

executing a substitute program segment associated with the corresponding substitute program segment in place of an original program segment associated with the exception inducing code.

14. (Original) The method of claim 13 further comprising second determining, by reference to an exclusion/inclusion policy associated with the instance of the exception inducing code, whether to perform the substitute program segment in place of the original program segment, wherein the exclusion/inclusion policy identifies at least a condition under which the corresponding substitute program segment will not be executed in place of an associated original program segment.

15. (Original) The method of claim 14 wherein the corresponding substitute program segment entry includes a portion of the original program segment replaced by the exception inducing code, and further comprising the step of determining that the condition under which the corresponding substitute program segment will not be executed has been fulfilled, and in response restoring within a program code execution stream the portion of the original program segment replaced by the exception inducing code.

16. (Original) The method of claim 13 wherein the substitute program segments comprise functions.

In re Application of DURR, et al.
Application No. 09/801,951

17. (Original) The method of claim 16 wherein the functions comprise application program interfaces.

18. (Original) The method of claim 17 wherein the application program interfaces are part of an operating system.

19. (Original) The method of claim 13 wherein the corresponding substitute program segment entry includes a called original program segment reference and a substitute program segment reference.

20. (Currently Amended) A computer-readable medium having computer-executable instructions for facilitating loading an executable program and associated modification code segments, the computer-executable instructions facilitating performing the steps of:

receiving a set of original program segments referenced by the executable program, each of the set of original program segments comprising executable binary code;

identifying a set of substitute program segments associated with ones of the original set of program segments;

modifying executable binary code of the ones of the original set of program segments to include an exception inducing code, the exception inducing code comprising executable binary code; and

creating an association between ones of the set of original program segments and ones of the set of substitute program segments.

21. (Original) The computer-readable medium of claim 20 further comprising computer-executable instructions facilitating:

determining a condition under which a particular substitute program segment will not be executed in place of an associated original program segment.

In re Application of DURR, et al.
Application No. 09/801,951

22. (Original) The computer-readable medium of claim 20 wherein the substitute program segments comprise functions.

23. (Original) The computer-readable medium of claim 22 wherein the functions comprise application program interfaces.

24. (Original) The computer-readable medium of claim 23 wherein the application program interfaces are part of an operating system.

25. (Original) The computer-readable medium of claim 20 wherein the creating an association step comprises building a table entry including a called original program segment reference and a substitute program segment reference.

26. (Original) The computer-readable medium of claim 25 wherein the called original program segment reference comprises an address.

27. (Original) The computer-readable medium of claim 25 further comprising computer-executable instructions facilitating:
storing a portion of the original program segment replaced by the exception inducing code in table entry.

28. (Original) The method of claim 25 further comprising computer-executable instructions facilitating:
storing a condition description specifying a program execution state under which a substitute program segment will not be executed in place of an associated called original program segment.

29. (Original) The computer-readable medium of claim 20 wherein the identifying step comprises searching a program modification database including a set of identified programs and corresponding substitute program segments.

In re Application of DURR, et al.
Application No. 09/801,951

30. (Original) The computer-readable medium of claim 20 further comprising computer-executable instructions facilitating:
loading an executable program into active process space;
identifying an in-memory patch within the program modification database corresponding to the executable program; and
inserting an in-memory patch within the executable program residing in the active process space.

31. *Amended.* (Original) A computer-readable medium having computer-executable instructions for facilitating implementing execution of substitute program segments associated with an active executable program, wherein the active executable program has been modified to include exception inducing code that induces an exception handled by an exception handling routine, the computer-executable instructions facilitating performing the steps of:

detecting an instance of the exception inducing code;
first determining a corresponding substitute program segment entry in a program segment substitution list; and
executing a substitute program segment associated with the corresponding substitute program segment in place of an original program segment associated with the exception inducing code.

32. (Original) The computer-readable medium of claim 31 further comprising computer-executable instructions facilitating:
second determining, by reference to an exclusion/inclusion policy associated with the instance of the exception inducing code, whether to perform the substitute program segment in place of the original program segment, wherein the exclusion/inclusion policy identifies at least a condition under which the corresponding substitute program segment will not be executed in place of an associated original program segment.

33. (Original) The computer-readable medium of claim 32 wherein the corresponding substitute program segment entry includes a portion of the original program

In re Application of DURR, et al.
Application No. 09/801,951

segment replaced by the exception inducing code, and further comprising computer-executable instructions facilitating:

determining that the condition under which the corresponding substitute program segment will not be executed has been fulfilled, and in response restoring within a program code execution stream the portion of the original program segment replaced by the exception inducing code.

34. (Original) The computer-readable medium of claim 32 wherein the substitute program segments comprise functions.

35. (Original) The computer-readable medium of claim 34 wherein the functions comprise application program interfaces.

36. (Original) The computer-readable medium of claim 35 wherein the application program interfaces are part of an operating system.

37. (Original) The computer-readable medium of claim 31 wherein the corresponding substitute program segment entry includes a called original program segment reference and a substitute program segment reference.

38. (Currently Amended) A program execution system facilitating modifying an executable program, including a set of original program segments referenced by the executable program, each of the set of original program segments comprising executable binary code, the program execution system comprising:

a program modification database for storing a set of substitute program segments associated with ones of the original set of program segments;

a program loader for installing a copy of the executable program into an active process space;

a program code modifier for altering, within the active process space, executable binary code of the ones of the original set of program segments to include an exception inducing code, the exception inducing code comprising executable binary code; and

In re Application of DURR, et al.
Application No. 09/801,951

a substitution list including entries describing an association between ones of the set of original program segments and ones of the set of substitute program segments.

39. (Original) The program execution system of claim 38 wherein the substitution list further comprises a policy description identifying a condition under which a particular substitute program segment will not be executed in place of an associated original program segment.

40. (Original) The program execution system of claim 38 wherein the substitute program segments comprise functions.

41. (Original) The program execution system of claim 40 wherein the functions comprise application program interfaces.

42. (Original) The program execution system of claim 41 wherein the application program interfaces are part of an operating system.

43. (Original) The program execution system of claim 38 wherein individual entries within the substitution list included a called original program segment reference and a substitute program segment reference.

44. (Original) The program execution system of claim 43 wherein individual entries within the substitution list include a portion of the original program segment replaced by the exception inducing code.

45. (Original) The program execution system of claim 38 further comprising:
an in-memory patch within the program modification database corresponding to the executable program; and
an in-memory patching function for inserting the in-memory patch within the executable program residing in the active process space.